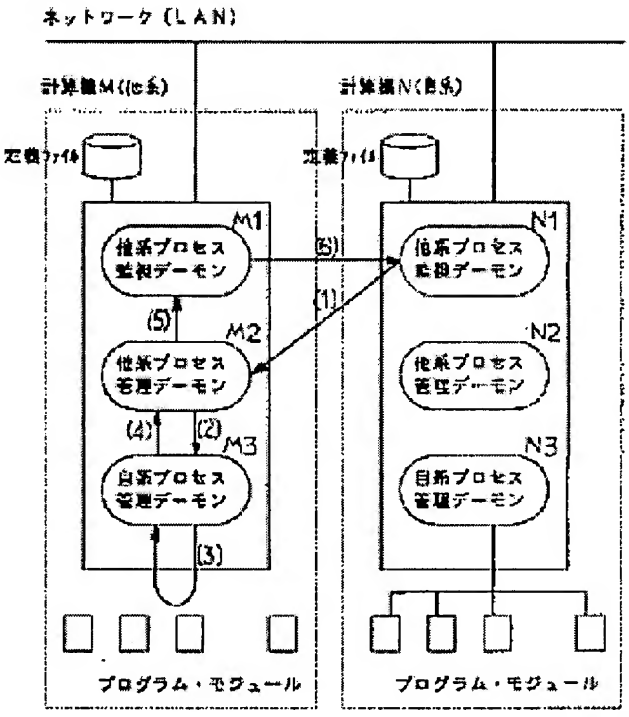
DISTRIBUTED PROCESS MANAGING SYSTEM

Publication number: JP9319720
Publication date: 1997-12-12
Inventor: SHIBATSUJI TOSHIO
Applicant: MITSUBISHI ELECTRIC CORP
Classification:
- international: ***G06F11/16; G06F11/30; G06F13/00; G06F15/16; G06F15/177;*** **G06F11/16; G06F11/30; G06F13/00; G06F15/16;** (IPC1-7): G06F15/16; G06F11/16; G06F11/30; G06F13/00
- European:
Application number: JP19960137987 19960531
Priority number(s): JP19960137987 19960531

Abstract of **JP9319720**

PROBLEM TO BE SOLVED: To confirm the request of information until the normal/abnormal finish of a program started within a computer and the sending of a signal within an issued server computer in the distributed process processing of plural computers. SOLUTION: The distributed process management of the plural computers connected by a network (LAN) is executed by three demon processors respectively provided with the functions of monitoring the process of the other system within each computer M1, N1, managing the process of the other system M2, N2 and controlling the process of its own system M3 and N3. As the result of this, each demon makes a speciality of each processing to reduce various event processings generated in the middle of processing to suppress the generation of a fault in terms of timing. In addition, history information of a process is centralized-controlled to recognize the load and the state of the computer of the other system by the computer of its own system.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平9－319720

(43)公開日　平成９年(1997)12月12日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ			技術表示箇所
Ｇ０６Ｆ	15/16	３７０		Ｇ０６Ｆ	15/16	３７０Ｎ	
	11/16	３１０			11/16	３１０Ｃ	
	11/30				11/30	Ａ	
	13/00	３５５			13/00	３５５	

審査請求　未請求　請求項の数12　ＯＬ　（全 11 頁）

(21)出願番号　特願平8－137987

(22)出願日　平成８年(1996) ５月31日

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号

(72)発明者　芝辻　俊男
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内

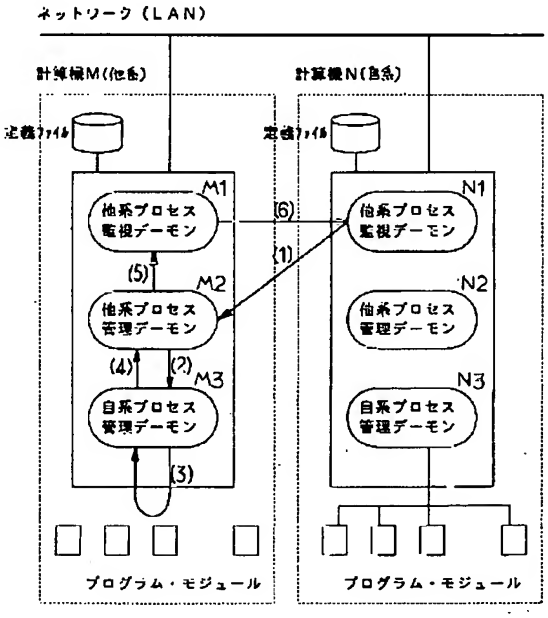
(74)代理人　弁理士　村上　博　（外１名）

(54)【発明の名称】　分散プロセス管理システム

(57)【要約】

【課題】　複数計算機の分散プロセス処理において、計算機内で起動したプログラムの正常／異常終了及びシグナルの送付までの情報の要求を、発行したサーバ計算機内で確認できるようにする。

【解決手段】　ネットワーク（ＬＡＮ）で結ばれた複数計算機の分散プロセス管理を、それぞれ各計算機内の他系プロセス監視Ｍ１，Ｎ１／他系プロセス管理Ｍ２，Ｎ２／自系プロセス管理Ｍ３，Ｎ３の機能を有する３つのデーモンプロセスにより行なわせる。その結果、各デーモンが各々の処理を専門に行なうことができ、処理の途中に発生する様々な事象処理を減らし、タイミング的な障害の発生を抑える。また、プロセスの履歴情報を集中管理し、他系計算機の負荷及び状態を自系計算機で認識する。



【特許請求の範囲】
【請求項１】　ネットワークを介して接続された複数計算機の分散プロセス管理システムにおいて、
自系計算機内のプログラムの動作を管理し、プログラムの起動，終了，同期，シグナルの送付等を要求する自系プロセス管理デーモンと、
他系計算機で動作する上記プログラムの起動，同期，シグナルの送付の要求を発行し、他系計算機で動作するプログラムの状態を監視する他系プロセス監視デーモンと、
他系計算機からの上記要求を受け取り上記自系プロセス管理デーモンへ送信し、その処理結果を自系計算機内の他系プロセス監視デーモンへ報告する他系プロセス管理デーモンの３つのデーモン構成からなる分散プロセス管理システム。
【請求項２】　上記デーモンは上記プロセスを管理する手段として定義ファイルを利用し、この定義ファイルは複数計算機間で動作するプロセス管理デーモンの計算機名を列挙していることを特徴とする請求項１記載の分散プロセス管理システム。
【請求項３】　上記定義ファイルで設定された計算機名は、自系計算機以外で起動するプログラムの起動要求発行先の他系計算機であって、その順序は優先起動順位を示し、上記要求発行先の計算機の状態をチェックし、計算機負荷に余裕がありかつプログラムの起動が可能な計算機を設定順に確認することを特徴とする請求項２記載の分散プロセス管理システム。
【請求項４】　上記定義ファイルには、予めプログラム起動権限を与える計算機名と起動パスを設定し、起動要求を行なう際にはプロセス名の指定のみの要求で、定義ファイル中に定義された計算機内の該当するパス下をサーチし、プロセス名と一致するモジュールが起動可能である場合、当該プロセスの起動を行なうことを特徴とする請求項２又は請求項３記載の分散プロセス管理システム。
【請求項５】　プロセス管理専用のライブラリをユーザに提供し、ユーザに提供するライブラリは、プログラムの起動（振り逃げ／完了），同期要求等を含み、上記デーモン間及びデーモン－アプリケーション間の通信にはメッセージを利用するとともに、計算機間をまたがる通信には、プロセス管理専用のソケット通信により行なうことを特徴とする請求項１から請求項４のいずれか１項に記載の分散プロセス管理システム。
【請求項６】　上記三種類のデーモンがプロセス管理のために利用する共用メモリを設置し、この共用メモリは他系プロセス情報管理テーブルと自系プロセス管理テーブルを有することを特徴とする請求項１から請求項５のいずれか１項に記載の分散プロセス管理システム。
【請求項７】　上記共用メモリの管理内容の更新時に当該内容が反映される共用記録媒体を設置し、上記共用メモリは他系計算機でプログラムが動作中に自系計算機が停止して再度動作可能となったタイミングで上記共用記録媒体から情報を取り込むことを特徴とする請求項６記載の分散プロセス管理システム。
【請求項８】　上記共用記憶媒体が複数計算機に接続されているシステムにおいて、各計算機内におけるプロセス管理動作履歴情報を上記共用記憶媒体に取り込み、現時点でプロセスの管理負荷の小さい計算機を見つけてその計算機に起動要求を行なうことを特徴とする請求項７記載の分散プロセス管理システム。
【請求項９】　他系計算機で動作するプロセスの負荷バランスを一定にするために、他系計算機でプロセスを起動する場合に、予め起動要求を行なう予定の計算機に現状プロセス管理状況を確認した後に要求を発行することを特徴とする請求項１から請求項８のいずれか１項に記載の分散プロセス管理システム。
【請求項１０】　他系計算機のプロセス起動要求を発行する場合、システム内に一計算機上のみにしか該当プロセスが存在しない場合、定義ファィルに設定された計算機順に起動要求を発行していくことを請求項１から請求項９のいずれか１項に記載の特徴とする分散プロセス管理システム。
【請求項１１】　上記他系プロセス監視デーモンはハートビートメッセージを定期的に発進し、起動要求先となっていた他系計算機が異常状態となった場合、当該他系計算機が発進するハートビートメッセージが一定時間途絶えることによって、他系計算機の異常と判断することを特徴とする請求項１から請求項１０のいずれか１項に記載の分散プロセス管理システム。
【請求項１２】　共用記憶媒体を持たない複数計算機システムにおいて、各計算機内におけるプロセス管理動作履歴をハートビートメッセージにデータとして乗せて発信することを特徴とする請求項１から請求項７ないし請求項９から請求項１１のいずれか１項に記載の分散プロセス管理システム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】この発明は、複数の計算機システムに存在するプログラムの状況管理に関するものである。
【０００２】
【従来の技術】近年、ネットワーク技術が進み分散プロセス処理システムの構成が容易にできるようになり、分散システム環境における遠隔操作技術に対する様々な発明がなされてきた。
【０００３】分散処理システムに関してのプロセス制御方法について、例えば特開平７－１６０６４６号に示されるものであるが、これはネットワークから与えられたタスクを受信する多重タスク処理ワークステーションのユーザが、ネットワーク・サーバから割り当てられたタ

スクに連結すること、及びそのタスクを一時的に又は恒久的に中断させるのを可能にするシステム及び方法である。

【０００４】また、特開平７－１３７７７号公報に示されるサーバプロセス管理装置では、対象装置内で動作するサーバプロセスを格納するサーバ動作条件情報格納部又は装置別に動作可能なサーバプロセスの動作条件を格納する装置別サーバ動作条件情報格納部と、動作しているサーバプロセスのアドレス情報を格納するプロセス情報格納部と、サーバプロセスを起動して前記プロセス情報格納部に起動したサーバプロセスの情報を格納するプロセス起動手段と、サーバプロセスを停止させて前記プロセス情報格納部から停止させたプロセスの情報を削除するプロセス停止手段と、動作中サーバプロセス情報を検索するプロセス情報検索手段と、サーバプロセスを起動要求したクライアントの情報を格納するクライアント情報格納部と、前記装置別サーバ動作条件情報格納部からクライアントプロセスから指定されたサーバの起動可能装置名及び動作条件を検索するサーバー情報検索手段と、クライアントプロセスまたはサーバプロセスからのプロセス情報削除要求を受け付けるプロセス情報削除要求受付手段と、サーバプロセスが停止したことを検出するプロセス監視手段を有するものである。

【０００５】

【発明が解決しようとする課題】しかしながら、従来の分散処理システムにおいては、サーバ側から要求のあったプロセスを、クライアント側で起動し停止させる処理方式において管理テーブルの構成に充填を置き、動作していたプロセスが正常終了することを前提としているものがほとんどであり、異常終了時に関しての動作について触れられていない問題点があった。

【０００６】この発明の目的は、純粋に計算機内で起動したプログラムの正常終了／異常終了及びシグナルの送付までの情報を、要求を発行したサーバ計算機内で確認できることを目的とする。

【０００７】

【課題を解決するための手段】請求項１の発明は、ネットワークで構成された複数計算機システムにおいて、自系計算機内のプログラムの動作を管理し、プログラムの起動，終了，同期，シグナルの送付等を要求する自系プロセス管理デーモンと、他系計算機で動作する上記プログラムの起動，同期，シグナルの送付の要求を発行し、他系計算機で動作するプログラムの状態を監視する他系プロセス監視デーモンと、他系計算機からの上記要求を受け取り上記自系プロセス管理デーモンへ送信し、その処理結果を自系計算機内の他系プロセス監視デーモンへ報告する他系プロセス管理デーモンの３つのデーモン構成からなる分散プロセス管理システムである。

【０００８】請求項２の発明は、プロセスを管理する手段として定義ファイルを利用する。定義ファイルには、複数計算機間で動作するプロセス管理デーモンの計算機名を列挙する様にする。

【０００９】請求項３の発明は、定義ファイルで設定された計算機の順序は、自系計算機以外で起動するプログラムの起動要求先の他系計算機の優先起動順位を示し、上記要求発行先の計算機の状態をチェックし、計算機負荷に余裕がありかつプログラムの起動が可能な計算機を設定順に確認する分散プロセス管理システムである。請求項４の発明は、定義ファイルに予め起動権限を与える計算機名と起動パスを設定しておくことにより、起動要求を行なう際には、プロセス名の指定のみの要求で、定義ファイル中に定義された計算機内の該当するパス下をサーチし、プロセス名と一致するモジュールが起動可能である時、そのプロセスの起動を行ない、以降そのプロセスの管理を開始するものである。

【００１０】請求項５の発明は、プロセス管理を実現する手段として、プロセス管理専用のライブラリをユーザに提供する。ユーザに提供するライブラリは、起動（振り逃げ／完了），同期要求等であり、デーモン間及びデーモン－アプリケーション間の通信にはメッセージを利用する。また、計算機間をまたがる通信には、プロセス管理専用Ｍａｃアドレスを利用したソケット通信でプロセスの管理を行なう。請求項６及び請求項７の発明は、プロセス管理が利用する管理テーブルとして共用メモリを使用し、分散プロセス管理を実現する三種類のデーモンが共通に参照でき、管理内容の更新時に共用記録媒体へ反映することにより、他系計算機でプログラムが動作中に自系計算機が停止して再度動作可能となったタイミングで、共用記録媒体から情報を取り込み、処理の連続が可能となる。

【００１１】請求項８の発明は、共用記憶媒体が複数計算機システムに接続されているシステムにおいて、各計算機内におけるプロセス管理動作履歴を置くことによって、他系計算機上のプロセス管理情報を共用記憶媒体から取り込み、現時点でプロセスの管理負荷の小さい計算機を見つけてその他系計算機に起動要求を行なう分散プロセス管理システムである。

【００１２】請求項９の発明は、他系計算機で動作するプロセスの負荷バランスを一定にする手段として、他系計算機でプロセスを起動する場合に、予め起動要求を行なう予定の計算機に現状プロセス管理状況を確認後に要求を発行するものである。

【００１３】請求項１０の発明は、他系計算機のプロセス起動要求を発行したが、システム内で一計算機上のみにしか、該当プロセスが存在しない場合、定義ファイルに設定された計算機順に起動要求を発行してゆき、必ず該当プロセスの起動を可能とする分散プロセス管理システムである。

【００１４】請求項１１の発明は、起動要求先となっていた他系計算機が異常状態となったとき、他系プロセス

監視デーモンが定期的に発進するハートビートメッセージが一定時間途絶えることによって、他系計算機の異常と判断する。異常と判断する時間は、ネットワーク負荷時を考慮した時間より遅くするものである。異常と判断した計算機に対して要求を行なっていたプロセスの情報を参照し、速やかにユーザーへ報告することができる。

【００１５】請求項１２の発明は、共用記憶媒体を持たない複数計算機システムにおいて、各計算機内におけるプロセス管理動作履歴をハートビートメッセージにデータとして乗せて発信するようにする。全系からのハートビートメッセージを受信することにより、各計算機毎の履歴情報を確認してプロセス管理状況負荷の一番低い計算機に対して起動要求を行なうものである。

【００１６】以上のように、この発明に係る複数計算機システムの分散プロセス管理システムは、自系計算機内のプログラムの管理を専用に行なう自系プロセス管理デーモンがプログラムの起動から終了を管理し、プログラムの終了ステータスを自系計算機の他系プロセス監視デーモンへ通知し、通知を受けた他系プロセス監視デーモンが、他系計算機で要求を行なった他系プロセス管理デーモンへプログラムの終了結果を報告し、報告を受けた他系プロセス管理デーモンが他系計算機の他系プロセス情報管理テーブルを更新し、共用記憶媒体に反映する。

【００１７】他系計算機上でプログラムを動作するのは、ユーザーアプリケーションプログラムが、プロセス管理専用ライブラリを利用し、自系計算機内の自系プロセス管理デーモンが、自系計算機で起動するものか、他系計算機上で起動するかの要求を判別し、自系計算機上のものである要求ならば、自系プロセス管理デーモンが、プログラムを起動し終了までを管理する。

【００１８】また、他系計算機上での起動要求であるならば、他系プロセス監視デーモンプロセスへ、その内容を通知し、他系プロセス監視デーモンが情報管理テーブルを構築及び更新し、他系計算機の他系プロセス管理デーモンへ通知する。

【００１９】また、この発明における情報管理テーブルは、シュアードメモリ上に他系プロセス監視デーモン，他系プロセス管理デーモン用，自系プロセス管理デーモンが共用で利用するテーブルを２面構成となる。

【００２０】つまり、他系プロセスの情報を管理するテーブルと自系プロセスの情報を管理する２つの情報テーブルにより管理される。

【００２１】また、この発明の他系プロセス管理テーブルの情報更新時に、システムで共用記憶媒体に情報管理テーブルをコピーすることにより、自系計算機にトラブルが発生し復旧した後でも他系計算機上で起動したプログラムの情報は確保されているので、処理の連続性が可能となる。

【００２２】

【発明の実施の形態】以下、この発明の実施の形態を図について説明する。

【００２３】実施の形態１．（分散処理システムのデーモン構成と処理）

図１は実施の形態１の分散プロセス管理システムを示す構成図であり、複数の計算機Ｍ，Ｎ，…がネットワーク（ＬＡＮ）を介して結ばれており、各計算機Ｍ，Ｎには、自系計算機内のプログラムの動作を管理し、プログラムの起動，終了，同期，シグナルの送付を要求する自系プロセス管理デーモンＭ３，Ｎ３と、他系計算機で動作するプログラムの起動，同期，シグナルの送付の要求を発行し、他系計算機で動作するプログラムの状態を監視する他系プロセス監視デーモンＭ１，Ｎ１と、他系計算機からの上記要求を受け取り、自系プロセス管理デーモンへ要求し、その処理結果を自系計算機内の他系プロセス管理デーモンへ報告する他系プロセス管理デーモンＭ２，Ｎ２、の３つのデーモンプロセスを備えている。

【００２４】次に、図１において実施の形態１の分散プロセスの処理の流れを説明する。まず、自系計算機Ｎの他系プロセス監視デーモンＮ１は他系計算機で動作するプログラムの起動，同期，シグナルの送付をソケット通信のプロセス起動要求（１）によって、他系計算機Ｍへ通知する。（１）の要求を受け取った計算機Ｍの他系プロセス管理デーモンＭ２は、その内容をメッセージ（２）によって自系プロセス管理デーモンＭ３へと通知する。メッセージ（２）を受け取った自系プロセス管理デーモンＭ３は、その要求に沿った処理を管理中のプログラム・モジュールに対して行なう（（３）；プロセスに対するアクション及び結果）。その処理結果をメッセージ（４）によって他系プロセス管理デーモンＭ２へと通知する。他系プロセス管理デーモンＭ２は、プロセス要求元の他系プロセス監視デーモンＮ１の計算機情報及び自系プロセス管理デーモンＭ３の処理結果をデータとして、メッセージ（５）によって他系プロセス監視デーモンＭ１へ通知する。メッセージ（５）のデータを受け取った他系プロセス監視デーモンＭ１はソケット通信（６）によって当該処理結果を他系プロセス監視デーモンＮ１に通知することにより、複数計算機間でのプロセス管理を可能とする。

【００２５】本実施の形態では、計算機内のデーモン間及びデーモンとアプリケーション間の通信にはメッセージを利用する。また、複数の計算機間をまたがるプロセス管理のための通信には、プロセス管理専用Ｍａｃアドレスを利用したソケット通信により行なうようにした。

【００２６】図１の分散プロセス管理システムによる効果は、３つのデーモン構成をとり、他系プロセス監視デーモン，他系プロセス管理デーモン，自系プロセス管理デーモンがそれぞれ専門の処理を担当することにより、処理時間の短縮とタイミングによる依存を排除する効果がある。

【００２７】実施の形態２．（定義ファイル）

実施の形態２は、プロセスを管理する手段として定義ファイルを利用する。定義ファイルには、複数計算機間で動作するプロセス管理デーモンの計算機名（自系計算機以外で起動するプログラムの起動要求発行先の計算機名）を列挙する様にする。

【００２８】具体的には、図３に示す定義ファイルを使用する。図３において、Ｍａｃ＿ｎａｍｅは分散プロセス管理を行なうマシン名（ＩＰアドレスで設定されたｈｏｓｔ　ｎａｍｅ）が全て列挙され、例えば図１の他系プロセス監視デーモンＮ１がプロセス要求（１）を発行する相手計算機として、このＭａｃ－ｎａｍｅパラメータの記述により決定する。Ｓｙｓｔｅｍパラメータは、分散プロセス管理を行なうネットワークに接続された計算機台数を設定するものである。Ｐｒｃ－ｃｏｕｎｔは、単一計算機内において管理可能なプロセス数を設定するものである。Ｐｒｃ－ｐａｔｈはプログラム起動の際に自系計算機内のプログラムを探すｐａｔｈを全て列挙したもので、Ｐｒｃ－ｐａｔｈによって予め設定されたパスにより、プログラムの起動要求はプログラム名のみで、自動的にパス下をサーチし、一番最初に見つけたプログラムモジュールを起動するようになる。Ｐｒｃ－Ｅｎｖは環境変数を設定するもので、起動要求のあったプログラム起動時に引き継がすものである。

【００２９】この実施の形態２の効果は、定義ファイルに予めプロセスを管理する計算機を設定することにより、ネットワークに繋がる対象外の計算機に対しては、要求を発行しないため、ネットワークの負荷を効率的になる。

【００３０】また、定義ファイルに予め複数のパス（ｐａｔｈ）を設定し、起動要求を行なう際にはプロセス名の指定のみの要求で、定義ファイル中に定義された計算機内の該当するパス下をサーチし、プロセス名と一致するプログラムモジュールが起動可能である時、当該プロセスの起動を行なうようにしたので、ユーザが指定するモジュール名のパラメータは名前だけで済み、モジュールを置くディレクトリも、定義されているパス（ｐａｔｈ）下であれば、各計算機毎に置け、起動要求プログラムは一つで済む効果がある。

【００３１】実施の形態３．（定義ファイルによる起動要求）

実施の形態３では、定義ファイルの起動権限を与える計算機名の列挙の仕方として、定義ファイルで設定された計算機名の順序をそのまま優先起動順位とする。そして、要求発行先の計算機の状態をチェックし、ＣＰＵ負荷に余裕があり、プログラムの起動が可能な計算機を設定順に確認し、プログラム起動が現在可能な計算機への要求を発行するようにする。

【００３２】例えば図２で示される複数計算機システムにおいて、図３のＭａｃ－ｎａｍｅパラメータによって設定された起動要求計算機順序にしたがって起動要求を行なう。すなわち、図３のＭａｃ－ｎａｍｅ　２Ａ＿３Ａ＿４Ｂ＿５Ｂ＿６Ｃ＿７Ｃ＿８Ｄの順序でプログラムの起動要求を行なうものである。

【００３３】まず、他系計算機２Ａに起動要求を発行し、タイムアウトもしくは、起動不可のリターンが戻された場合には、さらに計算機３Ａ→４Ｂ→５Ｂ→　→８Ｄと順次起動要求を行なう方式である。

【００３４】このＭａｃ－ｎａｍｅにより設定された起動要求計算機順序による効果は、起動要求を行なう計算機を限定しないが、必ず起動させなければならないモジュールを起動することができる。

【００３５】実施の形態４．（自系計算機からのプログラム起動の要求）

実施の形態４では、プロセス管理を実現する手段として、プロセス管理専用のライブラリをユーザに提供し、このライブラリを利用して、自系計算機からのプログラムの起動（振り逃げ／完了），同期要求等を行なう。

【００３６】図４はアプリケーションからのプログラム起動等の要求に対する処理の流れを示した図である。図において、Ｋ４はプログラムの起動，同期，シグナルの送付を要求するユーザアプリケーションであり、プログラムに対しての要求はライブラリによって行なわれる。

【００３７】ユーザアプリケーションＫ４はライブラリを介してプログラムの要求をメッセージ（４１）により自系プロセス管理デーモンＫ３へ要求する。その要求を受けた自系プロセス管理デーモンＫ３は、受け取った要求が自系計算機のプログラムに対するものか、他系計算機上のプログラムに対するものかを判別する（図４のＳＴＥＰ１００参照）。

【００３８】メッセージ（４１）の内容が他系計算機に対してのものであるならば、要求内容をメッセージ（４２－１）により他系プロセス監視デーモンＫ１へ通知し、他系プロセス監視デーモンＫ１は、図１で記したようにソケット通信（４２－３）により他系計算機へ上記要求を送信し処理動作を行なうこととなる。

【００３９】また、メッセージ（４１）の内容が自系計算機プログラムに対するものであるならば、自系プロセス管理デーモンＫ３は、（４１）の要求内容を元に自系計算機内部でのプログラム管理処理（４２－２）を行ない、その結果を待つ。

【００４０】他系計算機からの処理結果（４２－４）が他系プロセス管理デーモンＫ２を介してメッセージ（４３）で返されるか、自系計算機内部処理（４２－２）の結果メッセージが、自系プロセス管理デーモンＫ３に返された時、自系プロセス管理デーモンＫ３はライブラリを介して、その結果をユーザアプリケーションＫ４へリターン値として通知する。

【００４１】実施の形態５．（情報管理テーブル）

図５は本実施の形態の分散プロセス処理の情報管理方式を示す図であり、プロセス管理が利用する管理テーブル

として共用メモリを使用し、分散プロセス管理を実現する三種類のデーモンが共通して当該共用メモリを参照する。また、共用メモリの管理内容の更新時に共用記録媒体へ反映するようにし、他系計算機でプログラムが動作中に自系計算機が停止して再度動作可能となったタイミングで、共用記録媒体から情報を取り込む。その結果、自系計算機の状態の変化に影響を受けずプロセス処理の連続性を保つことができる。

【００４２】図５において、５０は分散プロセス管理する３つのデーモンが共通して書き込み／読み出すことができる共用メモリ（シェアードメモリ）であり、この共用メモリ５０は、他系プロセス情報管理テーブル５１と自系プロセス管理テーブル５２を備えている。

【００４３】他系プロセス情報管理テーブル５１は、初期起動時に図３の定義ファイルから他系計算機情報を取り込み構築され、他系計算機上で管理されるプログラムの計算機名，ソケット情報，要求開始時刻，プロセスＩＤ，親プロセスＩＤ，子プロセスＩＤ，プログラム名，ライブラリ発行履歴が管理される。また、自系プロセス情報管理テーブル５２には、自系プロセス管理デーモンＫ３が自系計算機上で管理するプログラムのプロセスＩＤ，起動時刻，親プロセスＩＤ，プログラム名，ライブラリ発行履歴，子プロセスＩＤが取込まれ、これら管理に必要な情報は予め図３の定義ファイルから初期起動時に構築されるものである。

【００４４】５３は自系計算機の異常時に影響を受けない共用記憶媒体であり、他系プロセス情報管理テーブルＫ５３が新たに他系計算機上のプログラムに対する要求及び報告を受け、このテーブル情報の更新時に、書き込み要求（５４）を介して共用記憶媒体５３に上記テーブル５１の内容が反映されるものである。

【００４５】この情報管理システムでは、プロセス管理が利用する管理テーブル（他系プロセス情報管理テーブル５１，自系プロセス情報管理テーブル５２）を備えた共用メモリ（シェアードメモリ）５０を使用し、分散プロセス管理を実現する三種類のデーモンが共通に参照でき、管理内容の更新時に共用記録媒体５３へ反映することにより、他系計算機でプログラムが動作中に自系計算機が停止して再度動作可能となったタイミングで、共用記録媒体から情報を取り込み、処理の連続を保持することができる。

【００４６】その結果、自系計算機に異常が発生し、復帰した場合、共用記録媒体５３に管理情報を持つことにより、他系計算機から情報が変更でき、その情報を元に処理の連続性が保持される効果がある。

【００４７】実施の形態６．（起動要求可否メッセージ）

ここでは、他系計算機で動作するプロセスの負荷バランスを一定にする手段として、他系計算機でプロセスを起動する場合に、予め起動要求を行なう予定の計算機に現状プロセス管理状況を確認後に要求を発行するようにする。

【００４８】図６において、Ｋは自系計算機、２Ａ～８Ｄは他系計算機であり、自系計算機Ｋから図３に示した定義ファイルを元に計算機優先順位の高い他系計算機２Ａに対して起動要求可否メッセージ（６－１）を送信する。他系計算機２Ａ上で管理が不可能である場合には応答メッセージ（６－２）によって否情報を返信する。返信メッセージ否情報を受けた自系計算機Ｋは、次に優先順位の高い他系計算機３Ａへ起動要求メッセージ（６－１）を送信し、応答メッセージ（６－２）を持つ。応答メッセージ（６－２）の情報が否である場合は、更に優先順位の次に高い計算機へ上記と同様の起動要求メッセージ（６－１）を送信し、応答メッセージ可情報が返信されるまで続ける。

【００４９】また、全ての他系計算機で応答メッセージ否が返信された場合は、ユーザにエラーリターンを返す処理を行なう。

【００５０】この起動要求メッセージ送信による効果は、起動要求を行なう予定の計算機に現状プロセス管理状況を確認した後、起動可であれば起動要求のあったモジュールを他系計算機上で動作させるようにしたので、各計算機内での必要以外のシステムコールの発行を抑止する効果がある。

【００５１】実施の形態７．（共用記憶媒体）

実施の形態７では、図５に述べた共用記憶媒体が複数計算機に接続されているシステムにおいて、その共用記憶媒体に各計算機内のプロセス管理動作履歴を置くことによって、他系計算機上のプロセス管理情報を共用記憶媒体から取り込み、現時点でプロセスの管理負荷の小さい計算機を見つけてその他系計算機に起動要求を行なうことを目的とする。

【００５２】図７において、７１は全ての計算機から参照可能な共用記憶媒体であり、この共用記憶媒体７１上に各計算機のプロセス管理履歴情報群のテーブル７２を設け、全ての計算機情報をどの計算機からも参照できるようにする。他系計算機２Ａ～８Ｄの情報は書き込み（７－１）によって各計算機上で管理情報の更新時に共用記憶媒体７１へ反映する。他系計算機上でプロセス起動を行なおうとする自系計算機Ｋは、要求を発行する前に共用記憶媒体７２の情報を読み取り（７－２）により得て、現在もっとも負荷の軽い計算機を選出する。

【００５３】このように、共用記憶媒体７１に各計算機内のプロセス管理動作履歴を置くことによって、共用記憶媒体７１を用いてシステム全体の管理履歴情報が参照でき、最も負荷の低い他系計算機へ直接要求が発行でき、起動要求から実行までの時間を短縮する効果がある。

【００５４】実施の形態８．（起動要求）

実施の形態８では、他系計算機のプロセス起動要求を発

行したが、複数計算機システム内で一の計算機上のみにしか該当プロセスが存在しない場合、定義ファイルに設定された計算機順に起動要求を発行して行き、必ず該当プロセスの起動を可能とすることを目的とする。

【００５５】図8において、２Ａ，３Ａ　８Ｄは他系計算機であり、自系計算機Ｋから起動要求ソケットメッセージを発行し、他系計算機上に要求されたプログラムが存在しない場合、自系計算機Ｋは定義ファイル中のＭａｃ－ｎａｍｅで指定されている全ての計算機に対して優先順序にしたがって要求を発行する。

【００５６】図8は他系計算機３Ａだけに目的とするプログラムモジュールが存在する場合を例示したもので、起動要求ソケットメッセージ（８１）を受けた他系計算機２Ａは、目的のプログラムモジュールが存在しない旨のソケット返信メッセージ（８１－２）を自系計算機Ｋに送信する。次に、自系計算機Ｋは起動要求ソケットメッセージ（８２）を計算機３Ａに発行し、計算機３Ａ側で該当プログラムモジュールを起動した後、ソケット返信メッセージ（８２－２）により自系計算機Ｋに起動報告がなされる。

【００５７】この実施の形態８による効果は、複数計算機システム内で一の計算機上のみにしか該当プロセスが存在しない場合、起動要求のあったロードモジュールに対して、必ず起動させることを目的とし、設定されている全ての計算機をサーチし、該当モジュールが必ず起動できることにある。

【００５８】実施の形態９．（ハートビートメッセージ）

実施の形態９では、起動要求先となっていた他系計算機が異常状態となった場合に備え、他系プロセス監視デーモンが定期的にハートビートメッセージを発進し、そのメッセージが一定時間途絶えた時、他系計算機の異常と判断するようにする。異常と判断する時間は、ネットワーク負荷時を考慮した時間より遅くするものである。この実施の形態によれば、異常と判断した計算機に対して要求を行なっていたプロセスの情報を参照し、速やかにユーザーへ報告することができる。

【００５９】この実施の形態９を図9により説明する。各計算機上の他系プロセス監視デーモンは、ハートビート（定期間隔通知）メッセージ９１を、定義ファイルに定義された全ての計算機に対して、自系計算機が正常であることを認識させるために発信する。通常、他系計算機に要求を発行した場合、処理の途中で計算機に異常が発生してもそれがわからなければ、処理中だと認識してしまうことになる。そこで定期連続メッセージ９１をハートビートの役割とし、９２に示すようにハートビートメッセージが断続的に途絶えた場合、メッセージが途絶えた計算機へ要求した処理を、即座に中断させるようにする。

【００６０】この実施の形態９による効果は、他系計算機で動作中のプロセスを自系で管理している場合に、他系計算機が異常となった時、その状態を即座に認識することができ、それに対応する処置を即座に行なうことができる。

【００６１】実施の形態１０．実施の形態１０は、図7に示す共用記憶媒体を持たない複数計算機システムにおいて、実施の形態７と実施の形態９の機能を実現するものであり、上記ハートビートメッセージ上に各計算機内におけるプロセス管理動作履歴をデータとして乗せて発信するようにする。計算機は全系からのハートビートメッセージを受信することにより、各計算機毎の履歴情報を確認してプロセス管理状況負荷の一番低い計算機に対して起動要求を行なうことができる。

【００６２】本実施の形態１０を図１０について説明する。この実施の形態では図7のような共用記憶媒体が存在しない。そこで、自系計算機Ｋは、ネットワークに接続された他の計算機より、各計算機のプロセス管理履歴情報を、定期的にハートビートの役割を兼ねたプロセス履歴情報メッセージ（２Ａ－１），…，（８Ｄ－１）により受信する。そして、そのメッセージを自系計算機Ｋ内のシェアードメモリ内テーブル上に（１０１）により書き込んで集計し、全体の履歴情報を計算機内で確認する。そして、上記履歴情報の読み取り（１０２）の結果から、他系プロセス監視デーモンは例えば一番負荷の軽い計算機８Ｄに対し起動要求ソケットメッセージ１０３を発行するものである。

【００６３】この実施の形態１０による効果は、共用記憶媒体を持たないシステムにおいても、ハートビートメッセージのデータに各計算機の管理履歴情報を乗せて送ることにより、共用記憶媒体を利用した時と同等のシステム全体の情報を各計算機で参照できるとともに、定期的なハートビートメッセージを利用するため、他系計算機の状態を一度に確認できる効果がある。

【００６４】

【発明の効果】以上のようにこの発明によれば、複数計算機システムにおける各計算機の負荷や正常状態以外の状態時でも、アプリケーションからの要求を、３つの分散プロセス管理を実現するデーモンプロセスにより、タイミング的な依存を抑えて実行できるように構成したので、予め提供する定義ファイルにより、システムの設定を行なうことにより、アプリケーションは各計算機情報を意識する必要がなく、ライブラリに定義するパラメータについてもシンプルな設定で済み、同アプリケーションのソースを修正することなく各計算機で実行できる効果がある。

【図面の簡単な説明】

【図１】　この発明の実施の形態による分散プロセス管理システムの構成及び処理の流れを表した図である。

【図２】　この発明の実施の形態に示すＬＡＮで接続された複数計算機システムを表した図である。

【図３】　この発明の実施の形態による複数計算機システムを管理するための定義ファイルの内容を示す図である。

【図４】　この発明の実施の形態によるアプリケーションから要求された自系計算機内での処理の流れを表わした図である。

【図５】　この発明の実施の形態による自系計算機の状態の変化に影響を受けず処理の連続性を保つ手段を表した図である。

【図６】　この発明の実施の形態による必要最小限のシステムコールの発行を実現するために、他系計算機に対する要求の前に予め伺いをたてて処理を行なうことを表す図である。

【図７】　この発明の実施の形態による共用記憶媒体を利用して、システム全体の管理情報を集中的に一ケ所に管理した処理を表わす図である。

【図８】　この発明の実施の形態による複数計算機システムにおいて、要求のあった該当するプログラムがシステム中に１台の計算機上にのみ存在する場合の処理の流れを表わした図である。

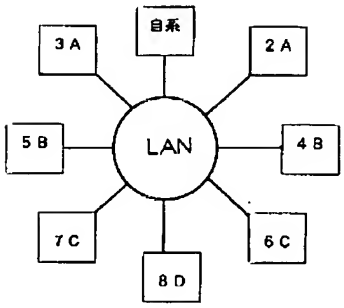
【図９】　この発明の実施の形態による計算機間の状態を確認する動作を表わした図である。

【図１０】　この発明の実施の形態による共用記憶媒体を利用しないシステムにおいて、計算機間の状態の認識と各計算機上の情報を組み合わせた処理を表わした図である。

【符号の説明】

Ｋ１，Ｍ１，Ｎ１　他系プロセス監視デーモン、Ｋ２，Ｍ２，Ｎ２　他系プロセス管理デーモン、Ｋ３，Ｍ３，Ｎ３　自系プロセス管理デーモン、Ｋ４　ユーザアプリケーション、（１）　Ｓｏｃｋｅｔ通信によるプロセス要求、（２）メッセージ、（３）　プロセスに対するアクション及び結果、（４）　結果報告メッセージ、（５）　プロセスに対する結果及び他系情報メッセージ、（６）結果報告Ｓｏｃｋｅｔ通信、２Ａ，３Ａ，４Ｂ，５Ｂ，６Ｃ，７Ｃ，８Ｄ　マシン名（ＩＰアドレスで設定されたＨｏｓｔ　ｎａｍｅ）、Ｓｙｓｔｅｍ　ＬＡＮに接続された分散プロセス管理を行なう計算機の台数、Ｍａｃ－ｎａｍｅ　マシン名（ＩＰアドレスで設定されたＨｏｓｔ　ｎａｍｅ）、Ｐｒｃ－ｃｏｕｎｔ各計算機毎の管理数、Ｐｒｃ－ｐａｔｈ　プログラム起動の際にプログラムを探すｐａｔｈ、Ｐｒｃ－Ｅｎｖ　プログラム起動の際に引継がせる環境変数、（４１）　メッセージ（ライブラリによる要求）、（４２－１）　他系計算機に対する要求メッセージ、（４２－２）　自系計算機内部でのプロセス管理処理、（４２－３），（４２－３）　Ｓｏｃｋｅｔ通信、（４３）　他系計算機でのプロセス管理処理結果メッセージ、（４４）　プロセス管理処理結果（ライブラリのリターン）、５０　共用メモリ、５１　他系プロセス情報管理テーブル、５２　自系プロセス情報管理テーブル、５３　共用記憶媒体、５４　共用媒体に対する書き込み要求、（６－１）　起動要求可否伺いメッセージ、（６－２）　可／否メッセージ情報、７１　共用記憶媒体、７２　プロセス管理履歴情報集中管理テーブル、（７－１）　書き込み、（７－２）　読み取り、（８１），（８２）　起動要求Ｓｏｃｋｅｔメッセージ、（８１－２），（８２－２）　要求処理結果Ｓｏｃｋｅｔ返信メッセージ、９１　ハートビート（定期間隔通知）メッセージ、９２　ハートビートメッセージの途断、１０１　シェアードメモリ内テーブルに書き込み、１０２　履歴情報読み取り、１０３　起動要求Ｓｏｃｋｅｔメッセージ、（２Ａ－１），（８Ｄ－１）　ハートビート＋プロセス履歴情報メッセージ。

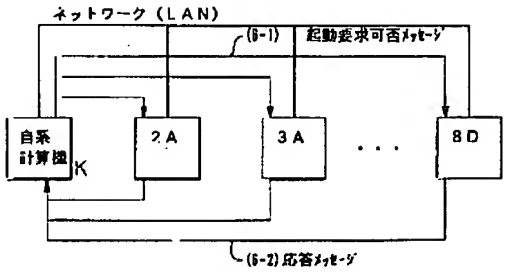
【図２】



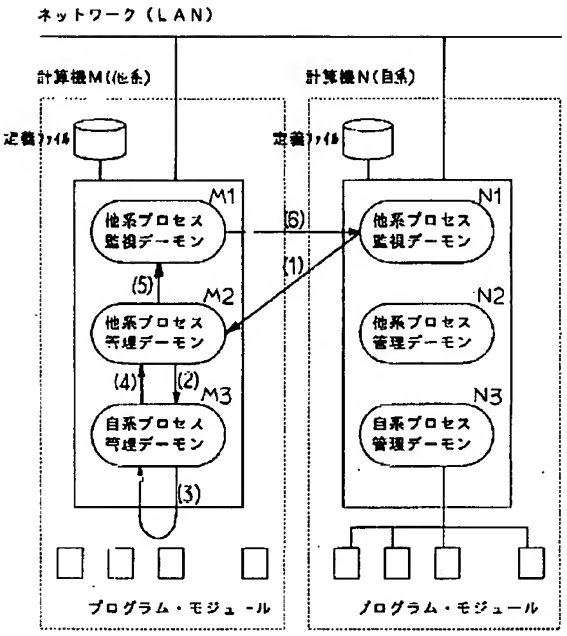
【図３】

```
System      7
Mac_name    2A_3A_4B_5B_6C_7C_8D
Prc_count   128
Pro_path    /etc_/usr/bin_/tmp
Prc_Env     TZ=JST-9
```

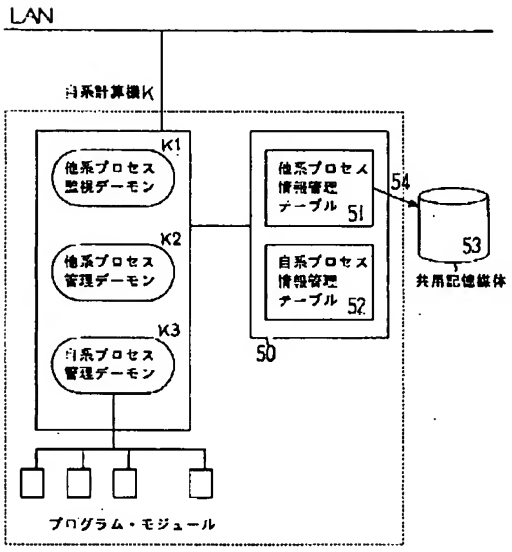
【図６】



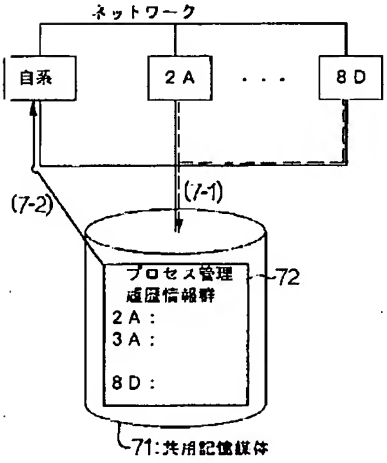
【図1】


ネットワーク（LAN）
計算機M(他系)
計算機N(自系)
定義ファイル
定義ファイル
M1
他系プロセス監視デーモン
M2
他系プロセス管理デーモン
M3
自系プロセス管理デーモン
N1
他系プロセス監視デーモン
N2
他系プロセス管理デーモン
N3
自系プロセス管理デーモン
(1)
(2)
(3)
(4)
(5)
(6)
プログラム・モジュール
プログラム・モジュール

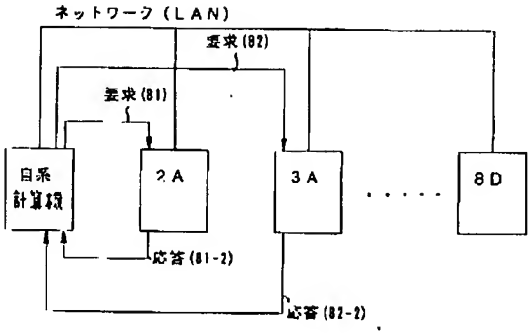
【図5】


LAN
自系計算機K
K1
他系プロセス監視デーモン
K2
他系プロセス管理デーモン
K3
自系プロセス管理デーモン
他系プロセス情報管理テーブル 51
自系プロセス情報管理テーブル 52
50
54
53
共用記憶媒体
プログラム・モジュール

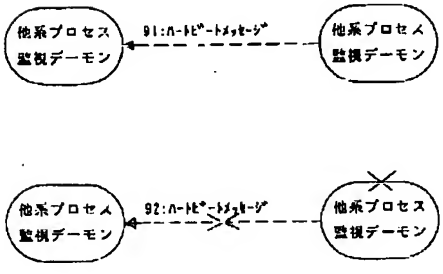
【図7】


ネットワーク
自系
2A
・・・
8D
(7-1)
(7-2)
プロセス管理
履歴情報群
2A：
3A：
8D：
72
71:共用記憶媒体

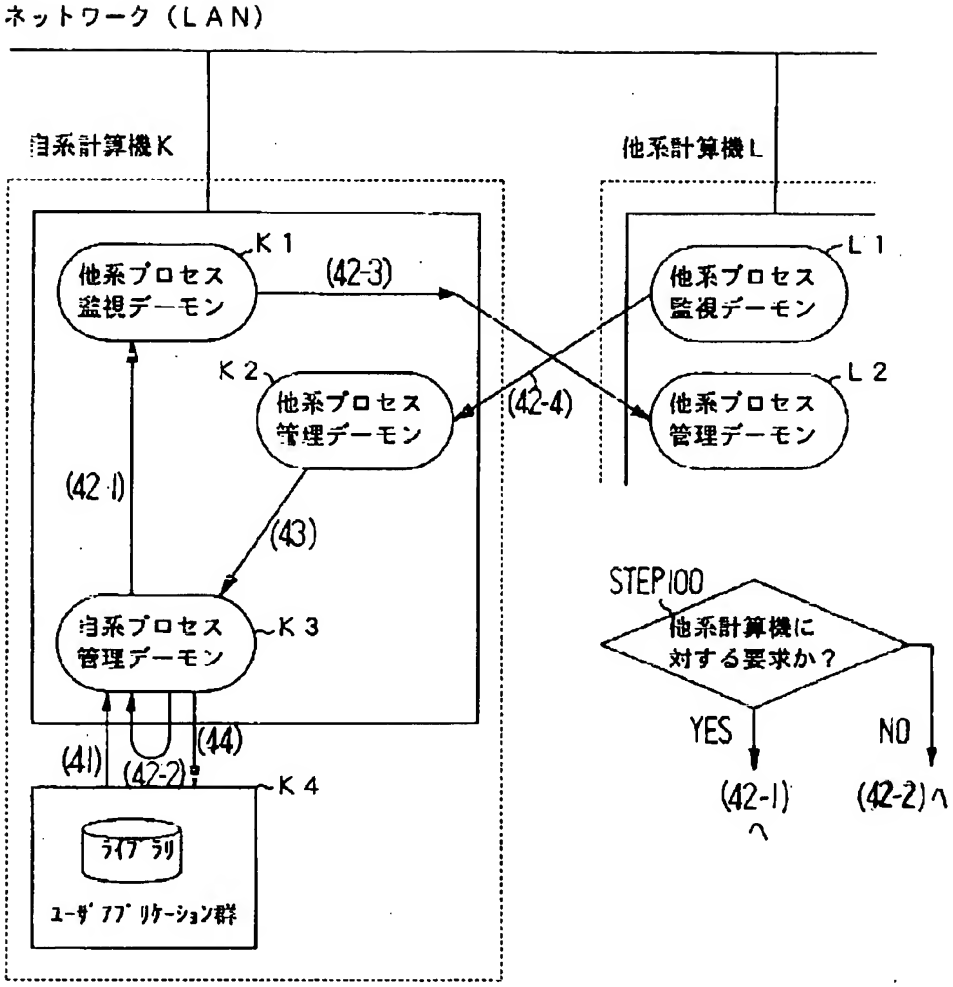
【図8】


ネットワーク（LAN）
要求(82)
要求(81)
自系計算機
2A
3A
・・・・・
8D
応答(81-2)
応答(82-2)

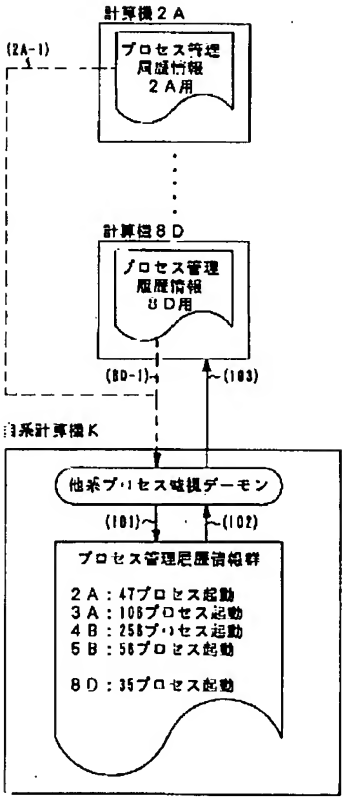
【図9】


他系プロセス監視デーモン
91:ハートビートメッセージ
他系プロセス監視デーモン
他系プロセス監視デーモン
92:ハートビートメッセージ
他系プロセス監視デーモン

【図4】


ネットワーク（LAN）
自系計算機K
他系計算機L
K1
他系プロセス
監視デーモン
(42-3)
L1
他系プロセス
監視デーモン
K2
他系プロセス
管理デーモン
(42-4)
L2
他系プロセス
管理デーモン
(42-1)
(43)
自系プロセス
管理デーモン
K3
(41)
(42-2)
(44)
K4
ライブラリ
ユーザアプリケーション群
STEP100
他系計算機に
対する要求か？
YES
NO
(42-1)へ
(42-2)へ

【図10】


計算機２Ａ
(2A-1)
プロセス管理
履歴情報
２Ａ用
計算機８Ｄ
プロセス管理
履歴情報
８Ｄ用
(8D-1)
(103)
他系計算機Ｋ
他系プロセス監視デーモン
(101)
(102)
プロセス管理履歴情報群
２Ａ：47プロセス起動
３Ａ：106プロセス起動
４Ｂ：256プロセス起動
５Ｂ：56プロセス起動
８Ｄ：35プロセス起動